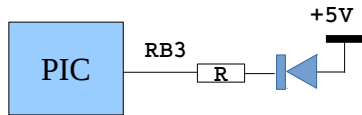


Corso di Laurea in  
Informatica  
Test di LAP1  
21/04/2016

Nome \_\_\_\_\_

Cognome \_\_\_\_\_

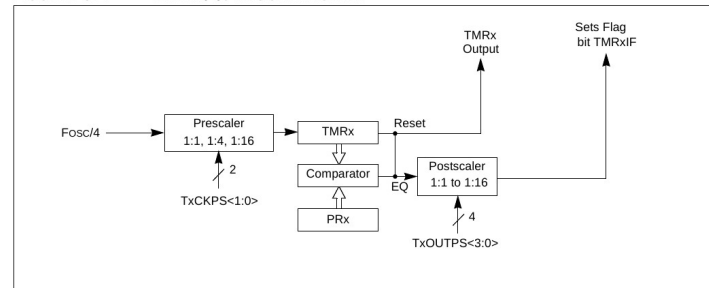
Matricola \_\_\_\_\_



A. Su un PIC18F, sia la porta RB3 collegata come in figura, quali delle seguenti impostazioni permette l'accensione del LED?

- ☐ `TRISBbits.TRISB3 = 0; LATBbits.LATB3 = 0;`
- ☐ `TRISBbits.TRISB3 = 0; LATBbits.LATB3 = 1;`
- ☐ `TRISBbits.TRISB3 = 1; LATBbits.LATB3 = 0;`
- ☐ `TRISBbits.TRISB3 = 1; LATBbits.LATB3 = 1;`

FIGURE 13-1: TIMER2/4/6 BLOCK DIAGRAM



B. Sia dato un PIC18F con  $F_{OSC} = 8 \text{ MHz}$ ; sia il TIMER2 (il cui schema è indicato in figura), configurato nel seguente modo:  $T2CKPS=0b10$ ,  $T2OUTPS=0b00$ ,  $PR2 = 0x83$ ; supposto che TMR2 sia inizialmente pari a  $0x22$ , dopo quanto tempo avverrà il prossimo interrupt?

C. Sia dato un PIC18F con  $F_{OSC} = 8 \text{ MHz}$ ; sia il TIMER2 configurato con un prescaler 1:4, postscaler 1:1, ed utilizzato per generare un PWM via software. Per il **Ton** si utilizza (per il registro PR2) il valore  $0x60$  e per **Toff** il valore  $0x80$ . Calcolare:

1. duty cycle, in percentuale
2. periodo complessivo del PWM

**D. Sulla base del seguente pezzo di codice , come si comporta il segnale su RB5?**

```
{
    int i;
    TRISBbits.TRISB5 = 0;
    for (;;) {
        LATBbits.LATB5 = 1 - LATBbits.LATB5;
        for (i = 0; i < 100*(LATBbits.LATB5+1); i++) ;
    }
}
```

- ☐ Rimane fisso ad un valore iniziale
- ☐ Cambia periodicamente stato, i due stati hanno lo durata differente, tuttavia tale durata non è determinabile
- ☐ Cambia periodicamente stato, i due stati hanno lo durata differente, e tali durate sono determinabili da un'analisi del codice
- ☐ Cambia periodicamente stato, i due stati hanno lo stessa durata, tuttavia tale durata non è determinabile

**E. Sia data la UART1, essa riceve un byte tuttavia, prima che tale byte venga “letto” dal software, la UART ne riceve un secondo. Cosa accade ai bit di stato?**

- ☐ RC1IF va a “0”
- ☐ RC1IE va a “1”
- ☐ OERR va “1” (overrun error)
- ☐ FERR va a “1” (framing error)

**F. Sia dato un ADC a 10 bit configurato da 0 a 5V; ad esso è connesso un sensore di distanza che genera una tensione secondo una legge lineare  $V = (\text{distanza in cm}) \cdot 0.03125$ . Determinare il valore atteso sull'ADC quando il sensore misura una distanza di 55cm.**

---

## Prova Pratica di LAP1

21/04/2016

Si realizzi un programma per microcontrollore 18F25K22 che presenti la funzionalita' di un **sistema di controllo per un pantografo XY** secondo le seguenti specifiche.

Mappa degli I/O:

- RA3: pulsante di **start**
- RA2: pulsante di **stop/emergenza**
- RC2: pulsante di **reset posizione**
- RB0: Accensione motore per spostamento **negativo** asse **X**
- RB1: Accensione motore per spostamento **positivo** asse **X**
- RB2: Accensione motore per spostamento **negativo** asse **Y**
- RB3: Accensione motore per spostamento **positivo** asse **Y**
- RB4: Target X raggiunto
- RB5: Target Y raggiunto
- AN0: specifica posizione target X, nell'intervallo [0, 1.023m]
- AN1: specifica posizione target Y, nell'intervallo [0, 1.023m]

1. La pressione del tasto **reset posizione** imposta le posizioni correnti X e Y a 0.
2. Ad ogni pressione del tasto **start**, il sistema determina le posizione desiderate X e Y, impostate tramite AN0 e AN1, e “sposta” i bracci del pantografo nella posizione desiderata, comandando opportunamente i motori tramite i segnali RB0, RB1, RB2 e RB3. Si suppone che i motori abbiano una **velocità costante di 0.2 m/s**. Si richiede che i motori siano attivati in contemporaneamente. La risoluzione desiderata, per gli spostamenti, è di **1 mm**. Al raggiungimento dei target il sistema accende i opportunamente i led su RB4 e RB5. Poiché la velocità è costante, i tempi di raggiungimento dei due target X e Y potranno essere differenti.
3. Qualora, durante il moto, si preme il tasto di **stop/emergenza**, la procedura dovrà essere arrestata; essa può essere riavviata premendo nuovamente il tasto **start**.
4. Durante l'operatività occorrerà visualizzare, tramite UART, le posizioni **target X e Y** e le posizioni **correnti X e Y**, in modo da verificare il corretto funzionamento del sistema.